# Multi-View Tracking of Multiple Targets with Dynamic Cameras

Till Kroeger[1], Ralf Dragon[1], Luc Van Gool[1,2]

[1] Computer Vision Laboratory, ETH Zurich
[2] ESAT - PSI / IBBT, K.U. Leuven
{kroegert, dragonr, vangool}@vision.ee.ethz.ch

**Abstract.** We propose a new tracking-by-detection algorithm for multiple targets from multiple dynamic, unlocalized and unconstrained cameras. In the past tracking has either been done with multiple static cameras, or single and stereo dynamic cameras. We register several moving cameras using a given 3D model from Structure from Motion (SfM), and initialize the tracking given the registration. The camera uncertainty estimate can be efficiently incorporated into a flow-network formulation for tracking. As this is a novel task in the tracking domain, we evaluate our method on a new challenging dataset for tracking with multiple moving cameras and show that our tracking method can effectively deal with independently moving cameras and camera registration noise.

## 1 Introduction

Simultaneous object tracking across multiple views is commonly solved with strong restrictive assumptions of known static cameras and planar movement constraints. It is easy to see that both constraints generally do not hold for many tracking tasks, such as simultaneous tracking using cameras on unmanned aerial vehicles, or tracking in synchronized dynamic videos from hand-held cameras. The knowledge about camera configurations will be unreliable or nonexistent. The movement on ground planes (GP) is an important special case, but even in standard tracking scenarios, e.g. pedestrians within cities, often too restrictive. We exploit connections between tracking and the Structure from Motion (SfM) domain, which can help making generic tracking scenarios solvable. Knowledge about camera localization and 3D structure can help with this task. Structure models built by today's SfM methods generally show good quality, are easy to create and widely available. Because of the ubiquity of available 3D models we propose to merge established methods for multi-view tracking-by-detection and localization methods developed for SfM to enable more generic tracking tasks.

**Our contribution** is a method for tracking-by-detection for multiple dynamic cameras, with known but noisy 6-DoF camera motion. We propose a novel method for linking data across time and views incorporating the motion uncertainty of the cameras.

We are the first to present a generalization of the strongly constrained 2D multi-view tracking scenario to unconstrained 3D scenarios. We do not make common restrictive assumptions, such as planar motion, known number of objects, constrained camera configurations, background modeling and explicit introduction of knowledge about the tracked objects movement characteristics in motion models.

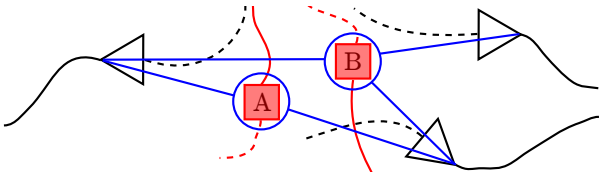


**Fig. 1.** Multi-view tracking from three dynamic cameras (colored in black, past: solid, future: dashed) of two moving objects (red). Object A is tracked in two, and B in three cameras. Camera locations are unknown and approximately determined by registration to SfM models.

We describe the tracking framework in § 2. In § 3 we explain unary and pairwise cost terms for the flow-network framework. In § 4 an experimental analysis is given. In § 5 we conclude and discuss future work. We use *view* and *camera* to refer to image data, orientation and position of one camera at a given time. We use *frame* to refer to image data from all views at a given time. A *dynamic camera* means a moving camera.

**Related Work:** Multi-object tracking has been studied extensively. The most successful methods define tracking as a global optimization over a complete sequence, with given frame-wise object hypotheses. This is usually called *tracking-by-detection*.

*Optimal single view data association* for many detections per frame has been formulated in [27,11]. [26] tracks without explicit detections on probabilistic occupancy grids. [2] extends this with iterations of discrete associations and continuous refinement.

*Data association across multiple views* adds the challenge of associating over time and views simultaneously. [15] solves one tracking graph for each view, multi-view couplings are solved in an additional graph. [9] solves associations across time using pre-computed merging candidates across views. [13] uses a homographic constraint to associate across views. [3,8,1] propose a probabilistic occupancy grid. Tracking is solved over discretized grid positions. These methods rely on static cameras. [16,6] propose multi-view tracking with dynamic cameras in a stereo setup.

*Data is usually collected* as independent detections from a given detector [27,20,2,1] [17,22,9,23] or by background subtraction [15,11]. [3,8,26,1,24] work directly on a discretized grid or volume representing the space of all possible locations.

*Several common constraints* are used to facilitate tracking. Motion is often limited to a known GP [3,26,2,8,3,17,22,9,13,24]. [16,6] assume planar motion, but infer the GP automatically. [26] uses global appearance constraints. [22] uses social grouping cues. [15,3,11,8] need static background and cameras. Specific camera configurations are needed: head-level cameras [8], top-down views [9], visible feet locations [13].

*Solutions* for association networks are found by Linear Programming (LP) [15,11,26] [1,20,9], Dynamic Programming (DP) [3,8,27], Energy minimization in MRF/CRFs [2,22]. To reduce computational demands greedy approximations are available [23].

*Camera registration/localization* has a variety of methods and distinct applications. In SLAM [5], methods estimate precise 6-DoF poses to 3D models, given location priors and feature matches. [19] solves robot self-localization given large-scale 3D models.

[25,10] propose methods for registering image collections for large scale models. [14] achieves high-precision registration for Videos to SfM models.

Our work is related to [9], with important differences. We perform the data association in 3D, while [9] uses a given GP. [9] has static entry/exit regions and hence the number of views to be used is known a priori. We incorporate camera and detector uncertainties into linking (16) and reconstruction probabilities (12), while [9] only uses these for error bounds while pruning reconstructions. [9] needs a form of enumeration for all possible sets of reconstructions, which grows factorially with number of cameras. Our work uses LP to optimally select the top reconstructions without enumeration, with a lower dependency on number of cameras, thus removing a serious bottlenecks of [9].

## 2 Tracking

We adopt a similar notation to [9]. We extract person detections using the Deformable Part Model detector [7]. Each 2D detection in camera $j$ at time $t$ is described by $\mathbf{x}_i^{(j,t)} = (x, s)$, with position $x$ and scale $s$. All detections at time $t$ from all cameras are denoted $\mathbf{X}^{(t)}$. One set of coupled detections from multiple cameras at time $t$ is denoted as reconstruction $\mathcal{R}_k^{(t)} \in \mathcal{R}^{(t)} \subset \mathcal{R}$. From each camera at most one detection can be included in each $\mathcal{R}_k^{(t)}$:

$$\mathcal{R}_k^{(t)} \subset \{\mathbf{X}^{(t)} | \forall \mathbf{x}_i^{(j,t)}, \mathbf{x}_{i'}^{(j',t)} \in \mathcal{R}_k^{(t)} : j \neq j' \vee i = i'\} \ . \tag{1}$$

Ideally one reconstruction $\mathcal{R}_k^{(t)}$ corresponds to one real world object which caused the detections in each view. We denote the number of detections in one reconstruction as its *cardinality*. The set $\mathcal{R}^{(t)}$ denotes all reconstructions at time $t$. A trajectory hypothesis is defined as $\mathcal{T}_u = \{\mathcal{R}_{u_1}^{(t)}, \ldots, \mathcal{R}_{u_n}^{(t+n)}\}$. This formulation allows missing detections for a real world object track in several views at different times. These missing detections can be due to detector errors or occlusions in a subset of views.

Similar in spirit to [9,27] we formulate multi-target tracking as a MAP problem and attempt to find an optimal set of tracks $\mathcal{T}$ which can be written as

$$\mathcal{T}^* = \arg\max_{\mathcal{T}} \ P(\mathcal{T}|\mathcal{R}) \quad = \quad \arg\max_{\mathcal{T}} \ P(\mathcal{R}|\mathcal{T})P(\mathcal{T}) \tag{2}$$

$$= \arg\max_{\mathcal{T}} \ \prod_i P(\mathcal{R}_i|\mathcal{T}) \prod_{\mathcal{T}_k \in \mathcal{T}} P(\mathcal{T}_k) \ . \tag{3}$$

This assumes conditional independence of likelihoods given $\mathcal{T}$ and independent object motion in (3). We further assume non-overlap of tracks, i.e. $\mathcal{T}_k \bigcap \mathcal{T}_j = \emptyset, \forall k \neq j$. The terms in (3) are defined as follows:

$$P(\mathcal{T}_k) = P_{en}(\mathcal{R}_{u_1}^{(t)}) \ P_{li}(\mathcal{R}_{u_2}^{(t+1)} | \mathcal{R}_{u_1}^{(t)}) \ \ldots \ P_{ex}(\mathcal{R}_{u_n}^{(t+n)}) \ , \tag{4}$$

$$P(\mathcal{R}_i|\mathcal{T}) = \begin{cases} P_{rec}, & \text{if } \exists \mathcal{T}_k \in \mathcal{T}, \mathcal{R}_i \in \mathcal{T}_k \\ 1 - P_{rec}, & \text{otherwise} \end{cases} \ . \tag{5}$$

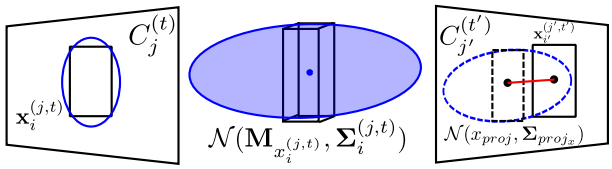


**Fig. 2.** Detection $\mathbf{x}_i^{(j,t)}$ from camera $C_j^{(t)}$ is transformed into 3D with center $\mathbf{M}_{x_i^{(j,t)}}$ and uncertainty $\mathbf{\Sigma}_i^{(j,t)}$. We propagate both into view $j'$ at time $t'$ and compute $D_3(x_{proj}, \mathbf{x}_{i'}^{(j',t')})$ (11) as the Mahalanobis distance in the image (colored in red).

This problem can be turned into a cost-flow network and solved *exactly* in an LP as in [9,1,26,11,15], or using DP [27,8,3], or *approximately* and fast as in [23]. We use an LP, similar to [9]. In order to solve the LP we have to compute a set of reconstruction hypothesis per frame $\mathcal{R}^{(t)}$ and provide probabilities $P_{en}$, $P_{ex}$, $P_{rec}$ for all reconstructions, and $P_{li}$ for all pairs of reconstructions. We turn the probabilities into costs $W_{en}$, $W_{ex}$, $W_{rec}$ and $W_{li}$ for the flow-network as in [27] (eq. 11), or [9] (eq. 13-16).

## 3 Modeling of Probabilities

The flow-network method links reconstructions across views and time given unary and binary costs. Similar to [27,9] we model these costs as probabilities $P_{rec}$ and $P_{li}$.

### 3.1 Extraction of Reconstruction Candidates

To start the tracking we need a set $\mathcal{R}^{(t)}$ of object hypotheses for all times $t$ containing the true solution. Because of this we focus on extracting the sets $\mathcal{R}^{(t)}$ large enough to ensure maximum recall. However, the combinatorial explosion prohibits inclusion of all feasible reconstructions in $\mathcal{R}^{(t)}$. We extract a set of $L$ top-ranking reconstructions for a each cardinality $m$ as follows: We apply a distance function $D_1(\mathbf{x}_i^{(j,t)}, \mathbf{x}_{i'}^{(j',t)})$ in 3D world coordinates between all pairs of two detections originating from different views, which we define in (9). Using this distance function we define the compactness of a reconstruction $\mathcal{R}_k^{(t)}$ as the sum of all pairwise detection distances $\sum_{x,x' \in \mathcal{R}_k^{(t)}, x \neq x'} D_1(\mathbf{x}, \mathbf{x}')$. This allows solving for the best (i.e. most compact) reconstruction of a given cardinality $m$ at time $t$ as an LP. We extract the $L$ top ranking solution using CPLEX for each cardinality and insert all extracted reconstructions at time $t$ into $\mathcal{R}^{(t)}$.

### 3.2 Localization of Detections and Reconstructions

Most previous multi-view multi-target tracking methods rely on the assumption of a given or automatically inferred ground plane (GP). However, GPs may not exist in some scenes, may not be visible (i.e. occluded in head-level cameras), or non-planar tracking is required. Localization of 2D object detections in 3D from head-level cameras using a GP assumption will be unreliable, even if a perfect GP is available, due to the small

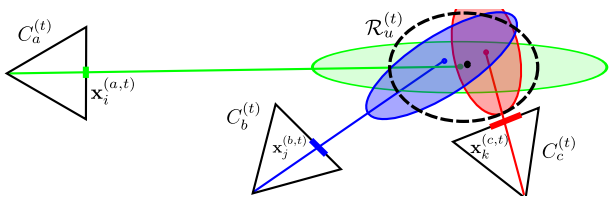


**Fig. 3.** A reconstruction $\mathcal{R}_u^{(t)}$ consisting of 3 detections (red, blue, green), seen from 3 cameras at time $t$ is localized in 3D. Detection uncertainties in 2D become depth uncertainties in 3D, which increase with distance to the camera. Eq. (8) gives center and uncertainty of $\mathcal{R}_u^{(t)}$ (black, dashed).

viewing angle. We side-step these problems by using the height of a detection as depth cue. Even if a bounding box is partly occluded, given the width and expected aspect ratio of a walking person, approximation of the height is possible. We use a distribution of heights with large uncertainty: mean $\lambda = 1.74\,\mathrm{m}$ and standard deviation $\sigma = \lambda/4$.

We assume a calibrated camera $j$ at time $t$ is given with approximately known pose $C_j^{(t)} = (R, tr)$, where $tr$ and $R$ denote the position and orientation given from some arbitrary noisy image-based registration method, and known internal calibration matrix $K_j$. Let $\mathbf{\Sigma}_{C_j^{(t)}}$ be the positional covariance of the camera. Given a detection $\mathbf{x}_i^{(j,t)} = (x, s)$ with center $x$ and scale $s$ in view $j$ at time $t$ we compute a 3D location and uncertainty in world coordinates for the detection. On undistorted images, the detection direction $\hat{x}$ is given by $\hat{x} = K_j^{-1}\tilde{x}$, where $\tilde{x}$ is homogeneous $x$ and $\|\hat{x}\| = 1$.

We compute the detection distance from the camera center as $d = \lambda \cdot f_j/s$, where $s$ is the pixel height of the detection in the image, and $f_j$ the camera's focal length. This is a reasonable distance approximation if the object is close to the optical axis. Knowing $d$, we model the 3D location in world coordinates as a normal distribution with mean $\mathbf{M}_{x_i^{(j,t)}}$ and covariance $\mathbf{\Sigma}_i^{(j,t)}$:

$$\mathbf{M}_{x_i^{(j,t)}} = R^T \cdot [(\hat{x} \cdot d) - tr] \;, \tag{6}$$

$$\mathbf{\Sigma}_i^{(j,t)} = R^T \cdot \mathrm{diag}\,(\lambda/2,\;\; \lambda/2,\;\; \lambda \cdot f_j/(4\,s)\,) \cdot R + \mathbf{\Sigma}_{C_j^{(t)}} \;. \tag{7}$$

The covariance is composed of two terms: Orthogonal to the optical axis, the detection 3D uncertainty is only dependent on the object scale $\lambda$. The detection height uncertainty in 2D becomes detection depth uncertainty in 3D as shown in figures 2 and 3. The second term is the isotropic camera localization uncertainty $\mathbf{\Sigma}_{C_j^{(t)}}$. A normal distribution over height in 2D will generally not translate into a normal distribution over depth in 3D. However, modeling (and thereby approximating) the true depth uncertainty as a normal distribution in 3D allows us to easily propagate the resulting uncertainty back into other 2D views and obtain a normal distributions again. See Fig. 2 and (10) for an explanation of the back-propagation.

To model the 3D reconstruction $\mathcal{R}_k^{(t)}$ consisting of many detections, we fit a normally distributed uncertainty with mean $\mathbf{M}_{R_k^{(t)}}$ and covariance $\mathbf{\Sigma}_{R_k^{(t)}}$ such that:

$$x \sim \mathcal{N}(\mathbf{M}_{R_k^{(t)}}, \mathbf{\Sigma}_{R_k^{(t)}}) \sim \frac{1}{Z} \sum_{x \in R_k^{(t)}} \mathcal{N}(\mathbf{M}_x, \mathbf{\Sigma}_x) \ . \tag{8}$$

Fig. 3 shows an example of 3 detections combined into one reconstruction. We use only one normal distribution for $\mathcal{R}_k$ and not the mixture of all distribution from included detections to allow quick propagation of a single normal distribution to different views.

### 3.3 Detection and Reconstruction Distances

In order to model $P_{rec}$ and $P_{li}$, or the reconstruction unary and pairwise costs, respectively, we need a set of geometric and appearance-based comparison measures between detections from different views and reconstructions. Given the localization and uncertainties of detections and reconstructions in 3D we will define four geometric and one appearance-based distance measure to be used for $P_{rec}$ and $P_{li}$, in §. 3.5 and 3.6.

- $D_1$: Mahalanobis distance between detections in 3D:

$$D_1(x, y) = D_{mah}(\mathbf{M}_y; \mathbf{M}_x, \mathbf{\Sigma}_x) + D_{mah}(\mathbf{M}_x; \mathbf{M}_y, \mathbf{\Sigma}_y) \ . \tag{9}$$

- $D_2$: Similar to $D_1$, Mahalanobis distance between two reconstructions $\mathcal{R}_k^{(t)}$,$\mathcal{R}_{k'}^{(t')}$ using mean $\mathbf{M}_{R_k^{(t)}}$,$\mathbf{M}_{R_{k'}^{(t')}}$ and covariance $\mathbf{\Sigma}_{R_k^{(t)}}$,$\mathbf{\Sigma}_{R_{k'}^{(t')}}$.
- $D_3$: Defined between a 2D detection $\mathbf{x}_i^{(j,t)}$, which has been projected into 3D as $\mathbf{M}_{x_i^{(j,t)}}$ with uncertainty $\mathbf{\Sigma}_i^{(j,t)}$, and a 2D detection $\mathbf{x}_{i'}^{(j',t')}$. (See Fig. 2). We project $\mathbf{M}_{x_i^{(j,t)}}$ into view $j'$ at time $t'$ with camera $C_{j'}^{(t')} = (R, tr)$ and produce $x_{proj} = K_{j'}[R\ tr] \cdot \mathbf{M}_{x_i^{(j,t)}}$. To propagate the uncertainty, the 3D covariance $\mathbf{\Sigma}_i^{(j,t)}$ is projected into view $j'$ at time $t'$ using the projection's Jacobian matrix $J^C(x)$ of $C_{j'}^{(t')}$. We also include $C_{j'}^{(t')}$'s localization uncertainty to handle camera errors:

$$\mathbf{\Sigma}_{proj_x} = J^C(x)^T \cdot (\mathbf{\Sigma}_i^{(j,t)} + \mathbf{\Sigma}_{C_{j'}^{(t')}}) \cdot J^C(x) \ . \tag{10}$$

  Using projected mean $x_{proj}$ and covariance $\mathbf{\Sigma}_{proj_x}$ we define $D_3$ as the Mahalanobis distance

$$D_3(x_{proj}, \mathbf{x}_{i'}^{(j',t')}) = D_{mah}(\mathbf{x}_{i'}^{(j',t')}; x_{proj}, \mathbf{\Sigma}_{proj_x}) \ . \tag{11}$$

- $D_4$: Defined between a reconstruction and a 2D detection equivalently to $D_3$ using the reconstruction's mean and covariance for reprojection.
- $D_5$: Between two 2D detections we define $D_5$ as the Earth mover's distance over RGB color histograms. We experimented with adding a HoG-based distance measure, but found it not to be helpful, due to large baselines between different views.

### 3.4 Entry and Exit Probability

Another often exploited advantage of static cameras is the possibility of defining explicit *entry* and *exit zones* in the images. Only in those areas are tracks allowed to start and end. This is easily modeled by setting $P_{en}$ and $P_{ex}$ to zero for all detections occurring outside these zones. For dynamic cameras this option does not exist. We estimate $P_{en} = P_{ex}$ for all reconstructions uniformly as the average missdetection probability of the detector over all sequences and cameras. These probabilities are transformed into costs for the flow-network as $W_{en} = W_{ex} = -\log(P_{ex}) = -\log(P_{en})$.

### 3.5 Reconstruction Probability

The probability $P_{rec}$ measures the pairwise similarity in appearance and spatial proximity of all detections in a reconstruction. For a reconstruction with detections of the same object, all detections should be similar in appearance and spatially close together. We set

$$P_{rec} = P(d_{rep}) \cdot P(d_{col}) \ . \tag{12}$$

$P(d_{rep})$ is computed as the probability that each detection reprojects well to all detections in other views. We average the distance $D_3$ for all pairs of included detections:

$$d_{rep} = \frac{1}{|\mathcal{R}_k^{(t)}|^2} \sum_{x_i, x_j \in \mathcal{R}_k^{(t)}} D_3(x_i, x_j) \ . \tag{13}$$

Average bounding box color dissimilarities are computed:

$$d_{col} = \frac{1}{|\mathcal{R}_k^{(t)}|^2} \sum_{x_i, x_j \in \mathcal{R}_k^{(t)}} D_5(x_i, x_j) \ . \tag{14}$$

The distributions of $d_{rep}$, $d_{col}$ are trained from ground truth and turned into matching probabilities $P(d_{rep})$, $P(d_{col})$. We transform the probability $P_{rec}$ into the node's flow cost $W_{rec}$ and add the detector confidence for every detection:

$$W_{rec} = \log\left(\frac{1 - P_{rec}}{P_{rec}}\right) + \sum_{x_i \in \mathcal{R}_k^{(t)}} \log\left(\frac{\beta(x_i)}{1 - \beta(x_i)}\right) \ , \tag{15}$$

where $\beta(x_i)$ describes the detection's false positive probability based on the detector score. Note that this cost neither penalizes nor prefers reconstructions with large or small cardinality. Unlike [9] we cannot assume to know the number of cameras in which an object is visible, and prefer to keep this score invariant to the cardinality.

### 3.6 Linking Probability

For two given reconstruction candidates $\mathcal{R}_k^{(t)}$, $\mathcal{R}_{k'}^{(t+1)}$ the transition probability $P_{li}$ indicates the probability of $\mathcal{R}_{k'}^{(t+1)}$ following $\mathcal{R}_k^{(t)}$ in a track. We set

$$P_{li} = P(d_{pro}) \cdot P(d_{col}) \cdot P(d_{cen}) \ . \tag{16}$$

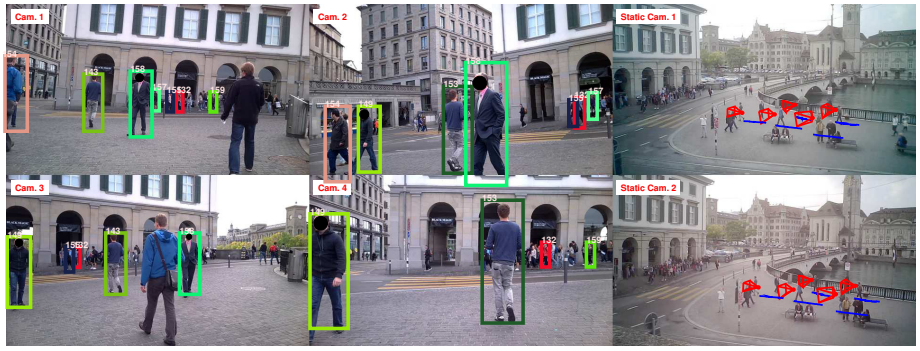


**Fig. 4.** Tracking result in 4 dynamic cameras of seq. 3. Static cameras 1+2 show current (red) and past (blue) estimated camera positions. Anonymized for publication purposes.

To establish these probabilities we will compute projection distances $d_{pro}$, pairwise color dissimilarities $d_{col}$, and the distance between the reprojections' center points $d_{cen}$. Let $m, n$ be the cardinalities of $\mathcal{R}_k^{(t)}$, $\mathcal{R}_{k'}^{(t+1)}$, respectively. We calculate how close the reconstruction $\mathcal{R}_k^{(t)}$'s 3D center projects to all detections $x_j \in \mathcal{R}_{k'}^{(t+1)}$ and vice versa:

$$d_{pro} = \frac{1}{n}\sum_{j=1}^{n} D_4(\mathbf{M}_{R_k^{(t)}}, x_j) + \frac{1}{m}\sum_{i=1}^{m} D_4(\mathbf{M}_{R_{k'}^{(t+1)}}, x_i) \ . \tag{17}$$

$d_{col}$ is computed as the average color dissimilarity between all detection pairs $x_i \in \mathcal{R}_k^{(t)}$, $x_j \in \mathcal{R}_{k'}^{(t+1)}$, similar to (14). We compute the distance between reconstructions centers $d_{cen} = D_2(\mathbf{M}_{R_k^{(t)}}, \mathbf{M}_{R_{k'}^{(t+1)}})$. This implicitly assumes maximum probability for stationary objects, which is generally not correct. However, this is an acceptable approximation because the frame-to-frame motion (1.5 meters/sec avg. walking speed) is 1-2 magnitudes smaller than the camera localization error. The distributions of $d_{pro}$, $d_{col}$ and $d_{cen}$ are trained from ground truth and turned into matching probabilities for (16). The probability is transformed into a cost as $W_{li} = -\log(P_{li})$.

## 4 Experiments

**Dataset for Evaluation:** We used three sets of videos for tracking, each set is called a *sequence*. Each sequence consists of 301 frames, seen from 7 synchronized cameras. Two of the 7 cameras are static, show the scene from a top-down view, are manually registered to the SfM model, and only used to create ground truth 3D locations for pedestrians and the 5 videographers. The remaining 5 cameras are dynamic, always seen from the static cameras, and used for tracking. In Fig. 4 and 5, 4 out of 5 dynamic and 2 static cameras are shown. Due to space constraints the 5th camera was omitted. Pedestrian tracks are annotated in all 7 views and identities across views are given. A manually determined GP is calculated for visualization and evaluation, but is not used in the algorithm. We manually collect 3D locations on this GP for all person tracks,

| | MOTA(%) | MOTP io(%) | MOTP 3d(m) | MT[%] | ML[%] | PT[%] | ids | Frag | C.Err1 (m) | C.Err2 (m) |
|---|---|---|---|---|---|---|---|---|---|---|
| S.1 | 1.01 | 0.88 / 0.93 | 2.65 / 1.36 | 7.14 | 28.57 | 64.29 | 21 | 91 | 0.73 / 0.65 | 2.16 / 1.79 |
| S.2 | 0.64 | 0.91 / 0.95 | 1.25 / 0.96 | 9.09 | 54.55 | 36.36 | 3 | 104 | 0.78 / 0.67 | 3.55 / 1.71 |
| S.3 | 1.44 | 0.89 / 0.94 | 1.32 / 0.88 | 0 | 54.55 | 45.45 | 4 | 119 | 0.62 / 0.48 | 2.82 / 1.21 |

**Table 1.** Tracking and video registration result for three sequences. Explanation in § 4.

including the videographers. The scene is not perfectly planar as can be seen in Fig. 4 and 5. Nevertheless, we expect the 3D localization error for all tracks to be below 30 cm.

**Video Registration:** For each sequence, we independently register all 5 dynamic videos to the SfM model using [21]. The SfM model covering the scene was manually created using high-resolution DSLR images. Because we have ground truth 3D position on a GP for each videographer track, we can evaluate the localization of each camera. The camera registration results in a mean and median error of $0.71$ and $0.62$ meters, respectively. Column *C.Err1* in table 1 show this positional error for all sequences separately. Column *C.Err2* shows the mean and median error between GT camera location and the estimated position of the corresponding videographer.

**Tracking:** We use an independently trained Deformable Part model detector [7] for person detection. Given registered videos we start the tracking by computing 3D positions and uncertainties for all detections, in all views and frames. We only have tracking annotation for persons which are seen from the two static cameras in at least one frame. We cannot evaluate the tracking for persons outside of this area. Therefore, we manually eliminate all detections, based on their 3D projection, which cannot be seen from the static cameras. This avoids creating tracks outside of the valid tracking region, which would result in a misleading FP number. Following § 3.1 we extract reconstruction candidates for all frames. We set $L = 100$ as sufficiently large set of hypotheses. We create the flow graph exactly as in [9], using our defined probabilities, and solve it using CPLEX. As evaluation criterion we use the metric presented in [18]: Identity switches (ids), Number of interruptions of a ground truth track (Frag), Mostly tracked (MT), Mostly lost (ML), Partly tracked (PT). Additionally, we use the CLEAR metric [4,12]: Multiple Object Tracking Accuracy (MOTA) and Multiple Object Tracking Precision (MOTP). The hit/miss decision is computed in image space. A hit has an intersection over union score of over $0.4$. We compute MOTP in image space as intersection over union (MOTP io, in percent, mean/median) and as the distance between estimated and true 3D track location (MOTP 3D, in meters, mean/median).

**Discussion:** We achieve consistently good precision in image space (MOTP io). In addition, we achieve good precision in 3D localization (MOTP 3d) considering that detection height was used as initial depth estimate and no GP was available. Additionally, we only have low confusion, as indicated by low identity switches (ids). The low ids score is partly explained by frequent interruptions of the ground truth track, resulting in high fragmentation (frag). Tracks break, rather than incorrectly switching targets. Often detections are incorrectly left out from a track, resulting in many false negatives and a low MOTA. This is due to the fact that the scene setup does not allow to specify a known cardinality for an object detection. Furthermore, several difficult tracks (severely occluded persons, far away persons) are missed altogether, resulting in a low MT, and

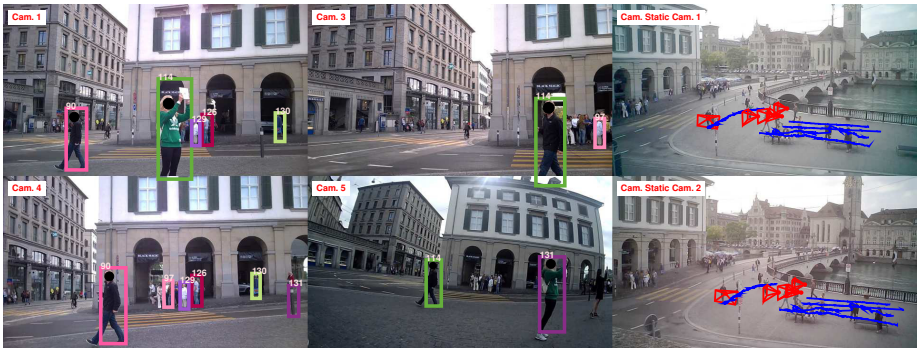


**Fig. 5.** Tracking result in 4 dynamic cameras of seq. 1. Static cameras 1+2 show current (red) and past (blue) estimated camera positions. Anonymized for publication purposes.

high ML. Fig. 4 and 5 show the tracking result from 4 of 5 dynamic cameras in two sequences. Identical colors and numbers in different views indicates the same track. The two static cameras show current estimated camera positions (red) and past videographer position (blue) on the GP. Typical failure cases include track 143 and 153 in Fig. 4: The object is correctly tracked but the tracks are not merged over views. Track 114 and 131 in Fig. 5 show a confusions between tracked objects within each track. In Fig. 5 some far-away persons are missed, due to unreliable detections.

## 5 Conclusion

With this work we contribute to the solution of multi-target tracking in multiple moving, approximately localized cameras. We extend an established tracking-by-detection framework using flow-networks, and show that even without many common constraints, such as availability of GP, static cameras or background subtraction, a satisfying tracking solution can be found nevertheless. This work presents the first generalization of multi-view multi-target tracking of objects on a GP to moving objects and cameras in 3D. Our tracking method is not limited to person detections. All objects with approximately constant size can be tracked given an appropriate detector. We consider several directions for future work. The MOTA scores we obtain are significantly lower than scores usually obtained on established datasets with static and known cameras, such as PETS2009. Primarily, this is due to the additional unknown and varying camera locations and the lack of available track entry and exit regions. We aim to improve this by using the 3D model visibility information to infer likely entry and exit regions. Another reason is the strong influence of incorrect and noisy camera poses. We aim to improve this by including pairwise essential matrix constraints and estimation of the relative poses directly within the tracking framework. We also explore possibilities of creating reconstruction candidates build from short 2D object tracklets, instead of single-frame detections to reduce computational demands.

**Acknowledgments:** This work was supported by the European Research Council (ERC) under the project VarCity (#273940).

## References


1. Andriyenko, A., Schindler, K.: Globally optimal multi-target tracking on a hexagonal lattice. ECCV (2010)
2. Andriyenko, A., Schindler, K., Roth, S.: Discrete-Continuous Optimization for Multi-Target Tracking. CVPR (2012)
3. Berclaz, J., Fleuret, F., Türetken, E., Fua, P.: Multiple Object Tracking using K-Shortest Paths Optimization. PAMI (2011)
4. Bernardin, K., Elbs, A., Stiefelhagen, R.: Multiple Object Tracking Performance Metrics and Evaluation in a Smart Room Environment Performance Metrics for Multiple Object Tracking. EURASIP (2008)
5. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: MonoSLAM: real-time single camera SLAM. PAMI (2007)
6. Ess, A., Leibe, B., Schindler, K., Van Gool, L.: Robust Multi-Person Tracking from a Mobile Platform. PAMI (2009)
7. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. PAMI (2010)
8. Fleuret, F., Berclaz, J., Lengagne, R., Fua, P.: Multi-Camera People Tracking with a Probabilistic Occupancy Map. PAMI (2008)
9. Hofmann, M., Wolf, D., Rigoll, G.: Hypergraphs for Joint Multi-View Reconstruction and Multi-Object Tracking. CVPR (2013)
10. Irschara, A., Zach, C., Frahm, J.M., Bischof, H.: From structure-from-motion point clouds to fast location recognition. CVPR (2009)
11. Jiang, H., Fels, S., Little, J.J.: A Linear Programming Approach for Multiple Object Tracking. CVPR (2007)
12. Kasturi, R., Goldgof, D., Soundararajan, P., Manohar, V., Garofolo, J., Bowers, R., Boonstra, M., Korzhova, V., Zhang, J.: Framework for performance evaluation of face, text, and vehicle detection and tracking in video: data, metrics, and protocol. PAMI (2009)
13. Khan, S.M., Shah, M.: A Multiview Approach to Tracking People in Crowded Scenes using a Planar Homography Constraint. ECCV (2006)
14. Kroeger, T., Van Gool, L.: Video Registration to SfM Models. ECCV (2014)
15. Leal-Taixé, L., Pons-Moll, G., Rosenhahn, B.: Branch-and-price global optimization for multi-view multi-target tracking. CVPR (2012)
16. Leibe, B., Cornelis, N., Cornelis, K., Van Gool, L.: Dynamic 3D Scene Analysis from a Moving Vehicle. CVPR (2007)
17. Leibe, B., Schindler, K., Van Gool, L.: Coupled detection and trajectory estimation for multi-object tracking. CVPR (2007)
18. Li, Y., Huang, C., Nevatia, R.: Learning to associate: HybridBoosted multi-target tracker for crowded scene. CVPR (2009)
19. Lim, H., Sinha, S.N., Cohen, M.F., Uyttendaele, M.: Real-time image-based 6-dof localization in large-scale environments. CVPR (2012)
20. Morefield, C.L.: Application of 0-1 Integer Programming to Multitarget Tracking Problems. Automatic Control (1977)
21. Moreno-Noguer, F., Lepetit, V., Fua, P.: Accurate Non-Iterative O(n) Solution to the PnP Problem. ICCV (2007)
22. Pellegrini, S., Ess, A., Van Gool, L.: Improving data association by Joint Modeling of Pedestrian Trajectories and Groupings. ECCV (2010)
23. Pirsiavash, H., Ramanan, D., Fowlkes, C.C.: Globally-Optimal Greedy Algorithms for Tracking a Variable Number of Objects. CVPR (2011)

24. Possegger, H., Sternig, S., Mauthner, T., Roth, P.M., Bischof, H.: Robust Real-Time Tracking of Multiple Objects by Volumetric Mass Densities. CVPR (2013)
25. Sattler, T., Leibe, B., Kobbelt, L.: Fast image-based localization using direct 2D-to-3D matching. ICCV (2011)
26. Shitrit, H.B., Berclaz, J., Fleuret, F., Fua, P.: Tracking Multiple People under Global Appearance Constraints. ICCV (2011)
27. Zhang, L., Li, Y., Nevatia, R.: Global Data Association for Multi-Object Tracking Using Network Flows. CVPR (2008)